

Quantizer를 위한 최적화된 Divider 구현

김재우, 조태현, 남기훈, 이광엽
서경대학교 컴퓨터공학과
전화 : 02-940-7240 / 핸드폰 : 018-339-6949

Design of an optimized Divider for a Quantizer

Jae-Woo Kim, Tae-Heon Cho, Ki-Hun Nam, Kwang-Youb Lee,
Dept. of Computer Engineering, Seokyeong University
E-mail : kim9128@dreamwiz.com

Abstract

본 논문에서는 현재 MPEG, JPEG 압축 알고리즘에서 쓰이는 DCT(Discrete Cosine Transform)기반의 손실 영상 압축에 사용되는 양자화(Quantization) 처리에 필요한 나눗셈 연산기를 제안한다. 영상 데이터 처리를 위한 양자화기(Quantizer)는 DCT로부터 매 사이클마다 영상 데이터를 입력 받아 양자화 처리를 해야 하며 보다 나은 영상 데이터를 위해 최종 나눗셈 결과 즉, 몫을 소수 첫째자리에서 반올림(Rounding)해야 한다. 이를 위해 반올림 동작이 추가된 Pipelined Nonrestoring Array Divider를 설계하였다.

제안된 방법의 타당성을 검증하기 위해 DCT로부터 나온 영상 데이터를 제안된 구조의 양자화기로 양자화하여 일반 양자화기에서 나온 압축된 데이터와 비교해 보았다. 또한 합성기(Synthesis)를 통하여 실제 하드웨어 크기를 분석하였다.

I. 서론

영상 압축의 필요성은 데이터의 대용량화와 저장, 전송 매체의 한계 때문에 영상 압축의 필요성이 대두

되고 있다. 영상 압축의 방식에는 데이터 손실의 존재 여부에 따라 무 손실 압축(Lossless Compression)과 손실 압축(Lossy Compression)으로 분류 할 수 있다. 손실 압축은 데이터의 손실이 있더라도 영상 데이터의 압축률을 향상하는 방법이다. 그렇기 때문에 손실 압축법이 무 손실 압축법에 비해 압축률이 더욱 좋다. 이런 데이터의 손실을 가하는 부분이 바로 양자화(Quantization)이다. 양자화가 가능한 이유는 인간의 눈은 영상 정보에 대하여 동일하게 민감하지 않으므로, 인간 시각특성에 의한 불필요한 부분이 발생한다. 그 결과 그러한 부분을 줄임으로서 영상 정보를 압축할 수가 있다. 양자화기(Quantizer)는 DCT로 구해진 화상정보의 계수값을 더 많은 '0'이 나오도록 일정한 값(Quantizer Value)으로 나오게 나누어준다. 따라서 영상 데이터의 손실이 있더라도 사람의 눈에서 이를 시각적으로 감지하기 힘들게 되므로 어느 정도의 데이터에 손실을 있더라도 압축률을 높일 수가 있는 것이다. 본 논문에서는 데이터의 손실이 있더라도 압축률이 좋은 손실 압축법에 필요한 Quantizer를 구현하기 위해 적합한 고속 Divider를 구현 하고자 한다. 영상 정보의 특성상 많은 데이터를 처리해야 하므로 그 결과 Quantizer를 위한 Divider 구현을 위해 Quantization Table 값을 이용하여 일반 Divide 연산보다 전반적인 처리 속도가 빠른 Pipelined Nonrestoring Array Divide 구조를 제안하였다.

본 논문은 IDEC 사업의 지원으로 작성되었습니다.

II. DCT 기반의 영상 압축 개요

2.1 DCT 기반의 영상 압축 기본 구조

DCT를 기반으로 하는 대표적인 영상 압축 표준이 JPEG이다. 아래 그림 1.은 JPEG 알고리즘의 부호 화기에 대한 대략적인 구성도를 나타내고 있다.

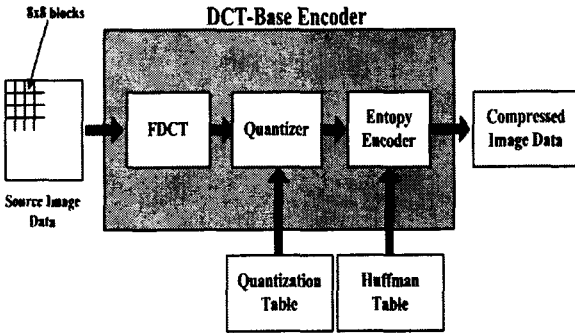


그림 1 .DCT기반의 영상 압축

DCT기반의 영상 압축 기반은 FDCT, 양자화기 그리고 엔트로피 부호화기(Entropy encoder)로 이루어져 있다.

2.2 FDCT(Forward Discrete Cosine Transform) 원리

DCT라는 것은 영상 정보를 주파수 별로 분해하는 것이다. 공간적인 영상 정보를 저주파에서 고주파 성분으로 변환함으로써 영상 정보를 압축하기 위한 준비 작업에 들어가는 것이다. 영상 정보를 분해하면 고주파와 저주파 성분으로 나누어지는데 이때 영상 정보에서 저주파라는 것은 이웃되는 픽셀들의 차이가 거의 없는 것을 의미한다.

보통 영상 정보의 경우에는 통계적으로 고주파보다는 저주파가 많기 때문에 화상을 주파수 별로 분해하면 값이 저주파에 물리게 되고 고주파에는 거의 존재하지 않아서 작은 계수 값을 갖는 고주파를 무시해도 영상에는 별로 영향을 미치지 않는다. 이와 같은 개념에 의하여 영상 정보의 압축을 수행하는 것이 DCT이다.[1]

이러한 DCT 이론을 바탕으로 2차원 8×8 FDCT 식을 표현하면 식(1) 과 같다.

$$F(u,v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

$$C(u), C(v) = \frac{1}{\sqrt{2}} \text{ for } u,v=0$$

$$C(u), C(v) = 1 \text{ otherwise} \quad (1)$$

2.3 Quantizer 원리

양자화의 목적은 더 좋은 압축률을 위한 것이며, 양자화 간격에 따라 압축률과 영상의 질이 결정된다. 양자화 간격은 고주파 영역으로 갈수록 그 간격이 더 커지는데 사람은 고주파 성분에 비교적 덜 민감하기 때문이다. 그 결과 8×8 FDCT를 적용한 후 구해진 64개의 FDCT계수는 미리 정해진 64개의 요소를 가진 양자화 테이블의 값에 따라 일정하게 양자화 과정을 거치게 된다. 식(2)는 양자화 과정을 식으로 표현한 것이다.

$$F^Q(u,v) = Round \left(\frac{F(u,v)}{Q(u,v)} \right) \quad (2)$$

식(2)에서 $F^Q(u,v)$ 는 양자화된 값을 나타내고 $F(u,v)$ 는 식(1)의 8×8 FDCT에서 나온 계수를 나타내고 $Q(u,v)$ 는 표 1.에 나타난 양자화 테이블 값을 나타낸다. 양자화 테이블은 JPEG에서 정한 표준 Luminance Table값을 사용하였다.[2]

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

표 1. Quantization table

식(2)에서와 같이 양자화의 원리는 8×8 FDCT를 거쳐서 나온 FDCT계수가 해당 Quantization Table의 값으로 나누어서 소수 첫째 자리에서 반올림 하는 원리이다. 그로 인하여 이 부분에서 실제적인 영상 정보의 손실이 발생 한다.

2.4 Entropy Encoder 원리

Entropy Encoder는 양자화기를 거쳐 나온 압축된 영상이 Zig-Zag Scan 전달 방법으로 Entropy Encoder에 전달 되서 데이터의 손실 없이 다시 한번

압축이 일어나는 단계이다. Entropy Encoder에는 RLE(Run Length Encoding)과 허프만 코딩(Huffman Coding)으로 구성 되어 있다.

III. Quantizer를 위한 Pipelined Nonrestoring Array Signed Divider

3.1 Nonrestoring Array Signed Divider 알고리즘

본 논문에서 사용한 기본 나눗셈 알고리즘은 Nonrestoring Array Signed Divider 구조를 사용하였다. Nonrestoring Array Signed Divider의 나눗셈 식은 식(3)과 같이 표현 된다.

$$s^{(j)} = 2s^{(j-1)} - q_{k-j}(2^k d) \quad \text{with } s^{(0)} = z \text{ and } s^{(k)} = 2^k s \quad (3)$$

Nonrestoring 나눗셈 알고리즘은 Restoring 나눗셈 알고리즘과는 다르게 Partial remainder값이 음수가 되더라도 나머지를 원래 값으로 복원 시키는 작업을 따로 하지 않고 다음 계산하는 단계에 계속해서 복원되지 않는 값을 반영 한다. 그 결과 Nonrestoring 나눗셈 알고리즘에서는 몫(Quotient)비트를 잘못 선택한 것을 보상 할 수 있도록 Restoring 방식과는 다르게 몫 비트에 음수의 값을 할당 할 수 있도록 하였다. 그 결과 Nonrestoring 나눗셈 알고리즘에서의 몫 비트는 1 또는 -1의 값을 할당한다. 즉 몫 비트가 1일 때는 덧셈 연산을 몫 비트가 -1일 경우에는 뺄셈 연산을 통하여 나눗셈 연산을 보정 할 수 있다. 식(4)는 Nonrestoring 나눗셈 알고리즘에서 몫을 할당하는 방법을 나타낸 식이다.

$$\begin{aligned} \text{if } 2s^{(j-1)} \geq 0 \text{ then } q_{-j} &= 1 \\ \text{if } 2s^{(j-1)} < 0 \text{ then } q_{-j} &= -1 \end{aligned} \quad (4)$$

이부분이 Nonrestoring Division이 Restoring Division 비해 2의 보수 표현으로의 확장이 용이하다는 것을 나타낸다. 즉 음수 표현이 용이하다는 것을 나타낸다. Quantization 연산은 Signed 나눗셈 연산이기 때문에 그로 인해 Quantization 연산에 필요한 Nonrestoring Array Signed Divider을 기본 Divider 구조로 사용하였다. 그림 2는 Nonrestoring 나눗셈의 Z-Z plot을 표

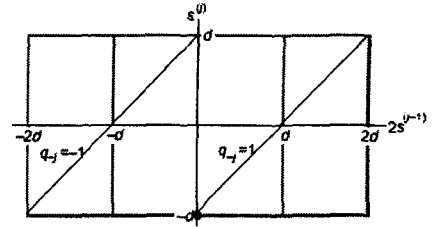


그림 2. Nonrestoring division Z-Z plot

현한 그림이다. 그림 3과 그림 4는 양자화기에서 사용한 기본 Nonrestoring Array Signed Divider 구조를 나타낸 그림이다.[3][4]

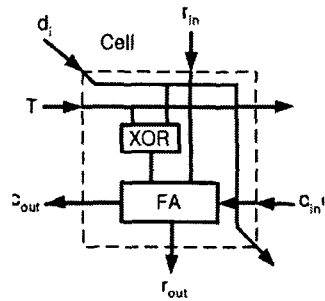


그림 3 .Controlled add/subtract (CAS) cell

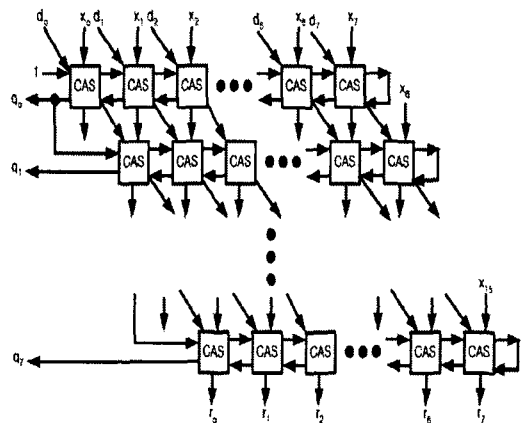


그림 4. 8 by 8 Nonrestoring Array Signed Divider

8 by 8 Nonrestoring Array Signed Divider의 각 가로열은 나눗셈 연산의 반복 연산을 덧셈과 뺄셈으로 펼쳐 놓은 것이다. 그리하여 각 가로열의 자리 올림수가 0이면 음수이므로 다음 열에서 덧셈을 수행하고 1이면 양수이므로 계속 뺄셈을 수행한다.

3.2 반올림 회로가 추가된 Pipelined Nonrestoring Array Signed Divider의 구현

Nonrestoring Array Signed Divider의 문제점은 처음 가로열에서 나온 자리 올림수가 최종 열까지 전달되어야 한다는 문제를 가지고 있다. 그 결과 $n \times n$ Divider는 총 n^2 의 지연이 걸린다. 그림 5.는 이런 문제를 보여주는 그림이다.

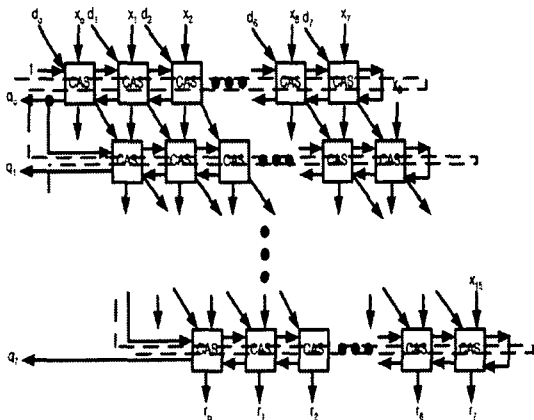


그림 5. Nonrestoring Array Signed Divider의 가장 긴 지연

이러한 문제를 해결하기 위해 Nonrestoring Array Signed Divider에 Pipeline기법을 사용하여 지연을 줄였다. 그림 6.은 Pipelined Nonrestoring Array Signed Divider를 구현한 그림이다.

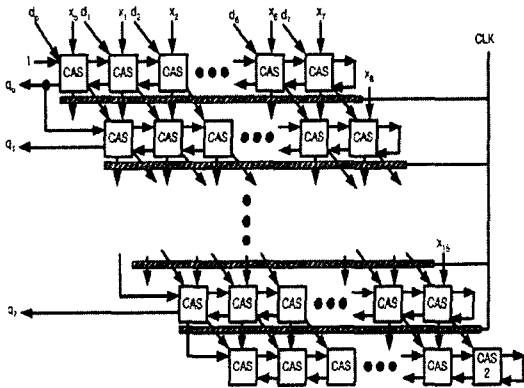


그림 6. Pipelined Nonrestoring Array Signed Divider 구조

그림 6.에서는 소수 첫째 자리까지 나눗셈을 구하기 위해 마지막 한단을 추가하였고 마지막 단의 CAS2는 전가산기가 아닌 반가산기로 구현 하였다. 또한 CAS와 CAS2의 크기를 줄이기 위해 전가산기와 반가산기를 XOR 게이트와 혼합시켜 CAS와 CAS2의 사이즈를 줄였다.

마지막으로 Pipelined Nonrestoring Array Signed와 반올림 회로가 추가된 회로가 그림 7.이다.

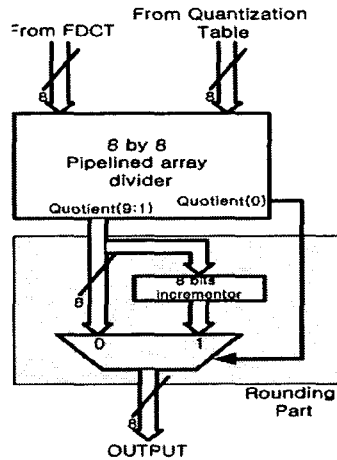


그림 7. Pipelined Nonrestoring Array Signed Divider와 반올림 회로가 추가된 회로

IV. 결론

본 논문에서는 양자화 연산에 필요한 나눗셈 연산을 하기 위해 8 by 8 Nonrestoring Array Signed Divider와 반올림 회로를 합하여 Pipeline을 수행하는 Divider를 구현하였다. 그리하여 DCT로부터 나온 영상 데이터를 제안된 구조의 양자화기로 양자화하여 일반 양자화기에서 나온 압축된 데이터와 비교해 보았다. 또한 합성기(Synthesis)를 통하여 실제 하드웨어 크기를 분석하였다.

참고문헌(또는 Reference)

- [1] Joan L. Mitchell, "JPEG" Kluwer Academic Publishers, 1992
- [2] Gregory K. Wallace, "The JPEG Still Picture Compression Standard", IEEE Transaction on Consumer Electronics, 1991
- [3] Behrooz Parhami, "Computer Arithmetic : Algorithms and Hardwarw Design", Oxford University Press, 2000
- [4] Israel Koren, "Computer Arithmetic Algorithms", John Wiley & Sons. Inc, 1978